CLAIMS

1. A method of enforcing policy in a multi-computer service application having a plurality of software modules that execute on a plurality of computers, the multi-computer service application further having access to a communications medium that allows data communications between different ones of the computers, the method comprising:

configuring logical output ports and logical input ports on different modules in accordance with a logical model of the multi-computer service application, wherein each logical input and output port is defined by port software;

configuring logical data connections between the logical output and input ports in accordance with the logical model;

configuring each port to communicate through different numbers of logical data connections without modifying the port software;

sending a notification from a particular module to a policy module;

the policy module responding to the notification by:

determining a request for one or more destination modules;

providing the request to an output port of the policy module;

the output port forwarding the request to input ports of a plurality of the modules in accordance with the configured logical data connections.

2. A method as recited in claim 1, wherein a particular output port is configurable during run-time to specify different logical data connections.

3. A method as recited in claim 1, wherein a particular output port is configurable during instantiation to specify different logical data connections.

4. A method as recited in claim 1, wherein the logical model of the multi-computer service application comprises model components, wherein each model component represents an abstract functional operation of the multi-computer service, the model components comprising hardware and software modules.

5. A method as recited in claim 4, wherein the model components have an associated blueprint that specifies the hardware and software modules represented by the model components.

6. A method as recited in claim 1, wherein the method further comprises:

monitoring, by the policy module, operation of the multi-service computer application during runtime; and

evaluating, by the policy module, the monitored operations against a policy.

7. A method as recited in claim 6:

wherein evaluating further comprises determining, by the policy module, a number of instances of each module used to implement the multi-computer service at any given time based on the policy; and the method further comprising responding from the policy module to changes in operating conditions by automatically specifying an action selected from a group of actions consisting of

deploying a new resource represented by a model component in the logical model, manipulating a module in multi-service computer application by sending requests to the module, and removing a module from the multi-service computer application.

8. A method as recited in claim 7, wherein the deploying comprises creating a physical instance of the model component, the logical input and output ports on the newly deployed resource being configured in accordance with logical connections specified in the logical model.

9. A computer-readable medium storing computer-executable instructions that, when executed on a computer, performs the method of claim 1.

10. A method of enforcing policy in a multi-computer service application having a plurality of software modules that execute on a plurality of computers, the multi-computer service application further having access to a communications medium that allows data communications between different ones of the computers, wherein the modules have logical input and output ports and logical data connections between modules, each logical port being defined by port software, the method comprising:

sending a notification from a particular module to a policy module;

the policy module responding to the notification by:

determining a request for one or more destination modules;

providing the request to an output port of the policy module;

23. A system to enforce a policy in a multi-computer service application having a plurality of software modules that execute on a plurality of computers, the multi-computer service application further having access to a communications medium that allows data communications between different ones of the computers, the system comprising:

a logical model of the multi-computer service application, the logical model having model components representing logical functions of the application;

a core runtime converter to create one or more module instances of the model components to implement logical functions represented by the model components, one of the module instances being a policy module, logical output ports and logical input ports on different modules being configured in accordance with the logical model, wherein each logical input and output port is defined by port software, logical data connections being configured between the logical output and input ports in accordance with the logical model, each port being configured to communicate through different numbers of logical data connections without modifying the port software; and

wherein, the policy module is configured to receive event notifications from a module instance, and in response to receiving an event notification, the policy module being further configured to:

(a) determine a request for one or more destination modules; and

(b) provide the request to an output port of the policy module, the output port being configured to forward the request to input ports of a plurality of the modules in accordance with the configured logical data connections.

11 10
24. A system as recited in claim 23, wherein a particular output port is configurable during run-time to specify different logical data connections, wherein the output port forwards the request to modules and input ports in accordance with the logical connections specified for said particular output port..

12 10
25. A system as recited in claim 23, wherein a particular output port is configurable during instantiation to specify different logical data connections, wherein the output port forwards the request to a plurality of modules and input ports in accordance with the logical connections specified for said particular output port.

13 10
26. A system as recited in claim 23, wherein the model components have an associated schema that specifies hardware and software modules represented by the model components.

14 10
27. A system as recited in claim 23, wherein the policy module is further configured to perform actions comprising:

monitoring operation of the multi-service computer application during runtime; and

evaluating the monitored operations against a policy.

28. A system as recited in claim 23, wherein to evaluate the monitored operations, the policy module determines a number of instances of each module used to implement the multi-computer service at any given time based on the policy; and

the policy module is further configured to:

respond to changes in operation conditions by automatically specifying an action selected from a group of actions consisting of deploying a new resource represented by a model component in the logical model, manipulating a module in multi-service computer application by sending events to the module, and removing a module from the multi-service computer application.

29. A system as recited in claim 28, wherein the policy module deploys the new resource by creating a physical instance of the model component, the logical input and output ports on the newly deployed resource being configured in accordance with logical connections specified in the logical model.

the output port forwarding the request to input ports of a plurality of the modules in accordance with the logical data connections.

11. A method as recited in claim 10, wherein a particular output port is configurable during run-time to specify different logical data connections.

12. A method as recited in claim 10, wherein a particular output port is configurable during instantiation to specify different logical data connections.

13. A method as recited in claim 10, wherein the logical model of the multi-computer service application comprises model components, wherein each model component represents an abstract functional operation of the multi-computer service application, the model components comprising hardware and software modules.

14. A method as recited in claim 13, wherein the model components have an associated blueprint that specifies the hardware and software modules represented by the model components

15. A method as recited in claim 10, wherein the method further comprises:

monitoring, by the policy module, operation of the multi-service computer application during runtime; and

evaluating, by the policy module, the monitored operations against a policy.

16. A method as recited in claim 15:

wherein the evaluating the policy module determines a number of instances of each module used to implement the multi-computer service at any given time based on the policy; and

the method further comprising:

responding, by the policy module, to changes in operation conditions by automatically specifying an action selected from a group of actions consisting of deploying a new resource represented by a model component in the logical model, manipulating a module in multi-service computer application by sending events to the module, and removing a module from the multi-service computer application.

17. A method as recited in claim 16, wherein the deploying comprises creating a physical instance of the model component, the logical input and output ports on the newly deployed resource being configured in accordance with logical connections specified in the logical model.

18. A computer-readable medium storing computer-executable instructions that, when executed on a computer, performs the method of claim 10.

19. A method comprising:

forming a scale-independent logical model of an application to be implemented by a distributed computer system, the model having multiple components representing logical functions of the application and intercommunication protocols;

converting the model components into one or more instances representative of physical resources that are used to implement the logical functions, the instances specifying communication ports on the physical resources and communication paths that link the physical resources; and

managing operation of the application by receiving notifications from certain instances on first communication ports and routing responses to the notifications to other instances on second communication ports.

20. A method as recited in claim 19, wherein the communication ports are configurable during run-time to specify different logical data connections.

21. A method as recited in claim 19, wherein the communication ports are configurable during instantiation to specify different logical data connections.

22. A method as recited in claim 19, wherein managing further comprises:

monitoring operation of the application during runtime; and

evaluating the monitored operations against a policy.